



Relatório Final – Iniciação Científica

Estágio científico e tecnológico I (EE015)

Estudo e Aplicação de Modelos de Previsão no Contexto de
Séries de Vazões de Rios

Submetido à
Faculdade de Engenharia Elétrica e de Computação (FEEC)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
CEP 13083-852, Campinas, São Paulo (SP)

Candidato: Daniel Neto
Orientador: Prof. Levy Boccato

1 Introdução

Uma série temporal é uma sequência de medidas feitas ao longo do tempo sobre um fenômeno de interesse. Em várias áreas do conhecimento, tais como engenharia, economia, matemática aplicada e computação, diversas séries temporais são particularmente relevantes não só pela perspectiva de análise e interpretação de seus históricos passados, mas também pela possibilidade de estimar seus valores futuros. Este último desafio dá origem ao problema de predição de séries temporais [1].

Um modelo genérico de previsão pode ser visto na Figura 1:

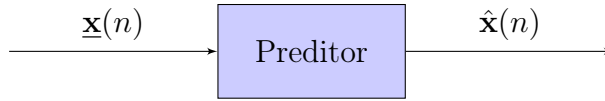


Figura 1: Estrutura geral de um preditor

O preditor recebe um vetor $\underline{\mathbf{x}}(n) = [x(n-L) \ x(n-L-1) \ \dots \ x(n-M-L+1)]^T$ de M amostras atrasadas da série temporal e produz uma saída $\hat{\mathbf{x}}(n)$, sendo L o passo de predição.

Um cenário em que o problema de predição de séries temporais aparece com destacada importância está associado a medidas de variáveis ambientais com comportamento aleatório, como vazões de rios, ventos e temperatura. Mais especificamente, no caso de séries de vazões de rios, tem-se interesse na previsão a fim de estabelecer um planejamento energético mais eficiente e seguro para uma determinada região. Isso se torna ainda mais relevante quando a principal fonte de geração de energia é constituída por usinas hidrelétricas, como é o caso do Brasil.

No entanto, existem alguns desafios envolvidos na previsão de uma série de vazões, tendo em vista sua natureza não-estacionária e a presença de componentes sazonais, as quais refletem os diferentes períodos de chuva/seca ao longo do tempo.

Este trabalho de iniciação científica tem como objetivo estudar e aplicar diferentes metodologias de previsão no âmbito de séries de vazões de rios brasileiros, de modo a criar um quadro comparativo entre as técnicas estudadas em um conjunto representativo de cenários.

Nesta primeira etapa do trabalho, foram estudados dois modelos clássicos da literatura, ambos de natureza linear: auto-regressivo (AR) e auto-regressivo de médias móveis (ARMA), detalhados nas seções 2.1 e 2.3 [1]. A descrição das características destes modelos é feita na Seção 2.

Um ponto que exigiu um estudo adicional está relacionado ao processo de adaptação do modelo ARMA. Uma vez que não há solução fechada, no sentido de mínimo erro quadrático médio (EQM), para os coeficientes do modelo, é necessário lançar mão de procedimentos iterativos de busca pelo vetor ótimo de parâmetros.

Neste trabalho, aproveitamos esta oportunidade para realizar um estudo de algoritmos de otimização inspirados no processo de evolução natural, especialmente os algoritmos genéticos e algumas técnicas baseadas na ideia de nicho ecológico. Estas abordagens heurísticas, ainda que não garantam a obtenção da solução ótima, despontam como alternativas promissoras para a busca por criarem um bom compromisso entre exploração do espaço de soluções candidatas e refinamento local. Além disso, possuem ampla aplicabilidade, uma vez que precisam apenas da informação de ordem zero da função sendo otimizada (no caso, do valor de erro quadrático médio de previsão sobre um conjunto de dados). A Seção 3 traz uma exposição de fundamentos de algoritmos evolutivos, em uma sequência lógica que parte do algoritmo genético básico até chegar em uma técnica mais sofisticada, denominada *clearing* (na Seção 3.3), com maior potencial de busca, especialmente em cenários multimodais.

Em seguida, foi feita a aplicação dos modelos AR e ARMA, com base em elementos da metodologia Box-Jenkins [1], à predição da série de Água Vermelha (descrita na Seção 4.1). Os resultados obtidos são apresentados e discutidos na Seção 4, concluindo, assim, esta primeira etapa da pesquisa.

2 Modelos de Previsão

2.1 AR

O modelo mais clássico de previsão é o auto-regressivo (AR, do inglês *auto-regressive*) [1, 2, 3]. No AR, o valor da série no instante n , aqui denotado por $x(n)$, é determinado a partir de uma combinação linear dos valores passados até um instante $n - M - L + 1$, onde M determina a ordem do modelo e L o passo de previsão (quantos instantes de tempo à frente pretende-se estimar). Em termos matemáticos, a regra de evolução temporal do modelo AR é dada por:

$$x(n) = a_1x(n - L) + \dots + a_Mx(n - M - L + 1) + \eta(n), \quad (1)$$

onde $a_i, i = 1, \dots, M$ são os coeficientes do modelo AR que ponderam as amostras passadas da série e $\eta(n)$ denota o erro instantâneo, cuja média é nula e cuja variância

(σ_η^2) é constante. Este último termo constitui um ruído branco (em inglês, *white noise*) e também é chamado na literatura de “choque aleatório” [1].

Considerando o caso em que $L = 1$, a expressão (1) pode ser colocada em uma forma mais compacta:

$$\sum_{k=0}^M w_k x(n-k) = \eta(n), \quad (2)$$

com $w_0 = 1$ e $w_k = -a_k$. Agora, é possível reconhecer o lado esquerdo em (2) como uma soma de convolução, o que nos permite pensar no modelo AR no contexto de sistemas lineares e invariantes com o tempo (LTI, do inglês *linear time-invariant*) [2].

Conforme mostrado em [2], dado um processo aleatório AR, ele pode ser gerado a partir da filtragem de um ruído branco $\eta(n)$ através de um sistema LTI cuja função de transferência no domínio \mathcal{Z} corresponde a:

$$H_{\text{síntese}}(z) = \frac{1}{\sum_{k=0}^M a_k z^{-k}} \quad (3)$$

Tendo em mãos uma série temporal $x(n)$, é necessário determinar os valores dos coeficientes $a_k, k = 1, \dots, M$ de tal modo que o modelo AR forneça a melhor aproximação possível para $x(n)$. Novamente, se considerarmos $L = 1$ para simplificação, multiplicarmos ambos os lados de (2) por $x(n-i)$, e aplicarmos o operador de esperança estatística, obtemos:

$$E \left\{ \sum_{k=0}^M a_k x(n-k)x(n-i) \right\} = E \{ \eta(n)x(n-i) \} \quad (4)$$

O termo à esquerda em (4) corresponde à autocorrelação do processo $x(n)$ para um atraso $(i-k)$, enquanto o termo à direita representa a correlação cruzada entre $x(n-i)$ e o ruído branco $\eta(n)$, a qual é nula para todo $i > 0$ [2]. Então, para qualquer $i > 0$, temos que:

$$\sum_{k=0}^M a_k r(i-k) = 0, \quad (5)$$

onde $r(m)$ é a autocorrelação do processo $x(n)$ para um atraso m . Considerando, finalmente, M valores sucessivos para o atraso $i > 0$, podemos montar um sistema de equações com M equações e M incógnitas, a saber, os coeficientes do modelo AR. Na forma matricial, temos que:

$$\underbrace{\begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r^*(1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(M-1) & r^*(M-2) & \dots & r(0) \end{bmatrix}}_{\mathbf{R}} \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix}}_{\mathbf{w}} = \underbrace{\begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(M) \end{bmatrix}}_{\mathbf{r}}, \quad (6)$$

onde $w_k = -a_k, k = 1, \dots, M$. O sistema em (6) define as chamadas equações de Yule-Walker, cuja solução (fechada) é dada por [1, 2]:

$$\mathbf{w} = \mathbf{R}^{-1}\mathbf{r}. \quad (7)$$

Interessantemente, a mesma solução para os coeficientes do modelo AR também pode ser obtida utilizando o arcabouço teórico de filtragem ótima (Wiener) [4, 2]. A vantagem desta última formulação é que ela modela explicitamente o problema como uma tarefa de otimização, na qual se deseja encontrar os coeficientes que minimizam a soma dos quadrados das diferenças (chamadas resíduos) entre o valor estimado e o valor correto. Esta solução também recebe o nome de Método dos Mínimos Quadrados [3].

Na prática, a fim de utilizar o modelo AR, é necessário definir o passo de predição L e o número de atrasos M . Então, a partir da série de interesse, constrói-se a matriz de autocorrelação \mathbf{R} e o vetor \mathbf{r} . Por fim, resolve-se a equação (7) para obter o conjunto ótimo de parâmetros \mathbf{w} .

2.2 MA

Outro modelo linear clássico na literatura de séries temporais assume que os valores de $x(n)$ resultam de uma combinação linear dos erros passados, ou, analogamente, de valores passados do próprio ruído branco, como mostra a expressão a seguir:

$$x(n) = \sum_{k=0}^K b_k \eta(n - k - J + 1), \quad (8)$$

onde $b_0 = 1$, J é o passo do ruído branco e K representa a ordem do modelo, o qual é conhecido como médias móveis (MA, do inglês *moving-average*) [2, 1].

Assim como ocorre no caso do modelo AR, o uso do MA requer a obtenção dos valores ótimos para os coeficientes que multiplicam os ruídos defasados. Entretanto, diferentemente do AR, não é mais possível determinar uma solução em forma fechada, de maneira que métodos iterativos e/ou heurísticos de busca devem ser empregados. Além disso, é necessário atentar-se à estabilidade do sistema, pois o acúmulo de erros pode levar à produção de saídas com energia (norma) divergente. Uma estratégia comumente utilizada para inibir a característica instável do sistema consiste em limitar a norma do vetor de parâmetros $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_K]^T$ [3].

2.3 ARMA

A combinação das ideias presentes nos modelos AR e MA dá origem a um modelo linear mais geral, denominado ARMA (do inglês, *auto-regressive moving-average*) [1, 2, 3]. Neste

caso, o valor da série temporal no instante n depende linearmente de valores passados da própria série e também do ruído branco, como mostra a expressão abaixo:

$$x(n) = \sum_{k=0}^M a_k x(n - k - L + 1) + \sum_{k=0}^K b_k \eta(n - k - J + 1), \quad (9)$$

onde as definições dadas para o modelo AR e MA ainda valem. É pertinente mencionar que o valor de L é comumente igual ao de J .

Devido ao uso implícito do MA no modelo ARMA, este último modelo herda a necessidade do uso de métodos iterativos e/ou heurísticos de busca para encontrar os valores ótimos dos coeficientes das regras lineares. Neste trabalho, segue-se um caminho semelhante ao trabalho de [5], recorrendo a meta-heurísticas de otimização inspiradas na natureza. Em particular, consideramos uma técnica de *niching*, denominada *clearing*, cuja descrição encontra-se na Seção 3.3. Além disso, à semelhança do modelo MA, a estabilidade do sistema é um aspecto que deve ser monitorado durante o processo de otimização dos parâmetros [3].

Embora o comportamento do modelo ARMA seja, de fato, mais geral, ele ainda realiza a estimativa do valor futuro da série com base em uma regra linear, o que pode ser uma limitação crítica quando se tenta capturar o comportamento temporal de séries mais complexas. Por isso, pretendemos estudar modelos mais flexíveis de previsão, por exemplo, baseados em redes neurais artificiais, durante a segunda etapa desta iniciação científica.

3 Algoritmos Evolutivos

A teoria da evolução por seleção natural, primeiramente formulada no livro de Darwin, *On the Origin of Species*, em 1859, é o processo pelo qual organismos tendem a uma maior aptidão com o passar do tempo como resultado da hereditariedade e mecanismos de seleção. Inspirada nestes princípios, surge a área conhecida como computação evolutiva, a qual utiliza o processo natural de evolução como base para desenvolver modelos computacionais capazes de resolver problemas [6, 7].

Uma das principais vertentes dentro da computação evolutiva está associada à elaboração de ferramentas para a solução de problemas de busca (ou otimização). Neste contexto, os algoritmos evolutivos tipicamente trabalham com múltiplas soluções candidatas, chamadas de indivíduos, em cada iteração, as quais podem gerar novas soluções por meio de mecanismos de reprodução e/ou mutação, e estão sujeitas ao processo de seleção natural, que determina quais soluções são mantidas na população tomando como base o nível de aptidão (*fitness*) das soluções com respeito ao objetivo do problema.

Os algoritmos evolutivos pertencem ao grupo das meta-heurísticas de otimização e, por isso, não oferecem garantias de obtenção da solução ótima, nem de convergência. Por outro lado, são metodologias gerais para o tratamento de problemas de otimização, exigindo apenas que seja possível calcular o valor da função objetivo (ou função de *fitness*) em qualquer ponto do espaço de busca. Além disso, sua estrutura básica pode ser facilmente adaptada de um problema para outro, e incorpora elementos que conferem ao algoritmo capacidade de explorar o espaço de busca – podendo, assim, escapar de ótimos locais –, e também de concentrar a busca nas regiões promissoras.

A estrutura básica de um algoritmo evolutivo é apresentada abaixo [6, 7]:

- I. **reprodução com herança:** os indivíduos têm a capacidade de gerar descendentes que herdam características de seus pais;
- II. **variação genética:** a população é submetida a possíveis mutações genéticas (geralmente não drásticas), as quais permitem uma maior exploração do espaço de busca;
- III. **seleção natural:** os indivíduos que avançarão para a próxima geração são selecionados de forma probabilística, sendo que a chance de serem selecionados é proporcional às suas respectivas medidas de aptidão (*fitness*).

Os tipos de algoritmos evolutivos existentes na literatura se diferenciam no uso e na ênfase que dão aos operadores supracitados. Neste trabalho, iniciamos o estudo pelos algoritmos genéticos para, então, estender sua estrutura incorporando técnicas baseadas no conceito de nichos ecológicos. A descrição feita neste relatório já particulariza as definições dos operadores para o contexto de um problema de otimização contínua, isto é, para o caso em que os parâmetros são números reais.

3.1 Algoritmo Genético Básico

Os algoritmos genéticos (em inglês, *Genetic Algorithms* (GAs)) foram propostos por J. Holland em 1975 [8], e são caracterizados pela estrutura básica detalhada na Seção anterior.

Em primeiro lugar, o algoritmo começa com uma população $\mathbf{P} \in \mathbb{R}^{N \times G}$ de N indivíduos gerados aleatoriamente no espaço de busca do problema (\mathbb{R}^G). A cada indivíduo, associamos um valor denominado *fitness*, o qual indica a sua qualidade como solução do problema. O vetor $F \in \mathbb{R}^{N \times 1}$ contém os valores de *fitness* dos indivíduos da população.

Em seguida, o algoritmo entra em um laço no qual as operações de cruzamento (*crossover*), mutação e seleção são sequencialmente aplicadas sobre a população, até que um critério de parada seja atingido.

3.1.1 Operador de recombinação

O *crossover* é dado da seguinte forma: percorre-se todos os indivíduos, os quais apresentam uma probabilidade p_c de participar de um cruzamento. Caso um indivíduo seja selecionado para *crossover*, seu material genético será combinado com o de outro indivíduo aleatoriamente escolhido da população.

Ambos os indivíduos geram juntos dois descendentes \mathbf{x}'_i e \mathbf{y}'_i , cujos parâmetros são combinações convexas dos dois genitores \mathbf{x}_i e \mathbf{y}_i . Este operador é conhecido na literatura como *crossover* aritmético [7], e tem como característica o fato de sempre gerar indivíduos factíveis se o espaço de busca for convexo. Em termos matemáticos, ele é empregado da seguinte forma:

$$\mathbf{x}'_i = a\mathbf{x}_i + (1 - a)\mathbf{y}_i \quad (10)$$

$$\mathbf{y}'_i = (1 - a)\mathbf{x}_i + (a)\mathbf{y}_i \quad (11)$$

sendo a um número aleatório retirado de uma distribuição uniforme entre 0 a 1. Após o estágio de recombinação, tem-se uma população intermediária baseada na união da população original com os novos descendentes gerados.

3.1.2 Operador de mutação

Sobre esta nova população, é aplicada a mutação. Percorre-se novamente cada um dos indivíduos, e caso ocorra a mutação (com probabilidade p_s), o indivíduo sofre uma perturbação aleatória em seu conteúdo genético. Uma forma típica de se realizar mutação para codificação real consiste em somar um vetor de variáveis aleatórias gaussianas de média nula e variância constante igual a σ_m^2 . Por conta disso, este operador recebe o nome de *mutação gaussiana* [6, 7]. Considerando um indivíduo definido por $\mathbf{x} = [x_1 \ \dots \ x_M]^T$, o resultado da mutação é dado por:

$$x'_i = x_i + m, \quad (12)$$

onde i denota a i -ésima característica do indivíduo selecionada para mutação e $m \sim N(0, 1)$.

3.1.3 Operador de seleção

A seleção deve privilegiar a existência dos indivíduos de maior *fitness* na próxima geração com o cuidado de preservar certa diversidade na população, ampliando as chances de explorar novas regiões do espaço de busca.

Um método clássico de seleção consiste em extrair N_t indivíduos da população aleatoriamente (com reposição) e selecionar aquele com maior *fitness* para a próxima geração. Por isso, o método recebe o nome de seleção por torneio e é repetido N vezes, de modo que a nova população mantenha o tamanho original.

Note que, quanto maior o valor de N_t , maior será a pressão seletiva, e, portanto, mais rápida a convergência ao ponto de maior *fitness* já identificado (na região explorada). De fato, mesmo que N_t seja pequeno, o algoritmo genético possui a tendência de acabar convergindo todos os indivíduos da população para a mesma região após um número suficiente de gerações.

Existe também a possibilidade do indivíduo de melhor *fitness* não prosseguir para a próxima geração após a aplicação do torneio. Caso isso ocorra, troca-se o indivíduo de menor *fitness* da nova população por aquele, de modo a não perdermos boas soluções.

3.2 *Niching*

Com o objetivo de tentar impedir a convergência de todos os indivíduos da população a uma única região no espaço de busca, uma característica intrínseca aos algoritmos genéticos clássicos, introduz-se o conceito de nichos ecológicos (em computação, *niching*) na abordagem do problema. O nicho ecológico é o conjunto de todas as tolerâncias e exigências de uma espécie, isto é, todos os recursos que permitem sua sobrevivência em dado ambiente. Como um nicho apresenta recursos limitados, um aumento no número de organismos no respectivo nicho leva a uma diminuição da prosperidade associada a ele.

No contexto computacional, diz-se que dois indivíduos pertencem a um mesmo nicho se eles ocupam uma mesma região do espaço de busca, isto é, se a distância (Euclidiana) entre eles é inferior a determinado limiar σ .

Neste caso, para evitar que o algoritmo tenda a concentrar todos os indivíduos na mesma região, inspirado no fato de que um indivíduo precisaria compartilhar recursos com outros de seu nicho, o *fitness* original (f_i) do i -ésimo indivíduo é reduzido por um fator que reflete o grau de aglomeração em torno dele no espaço de busca.

Em outras palavras, define-se um novo valor de *fitness* (f'_i), chamado de *fitness* compartilhado (do inglês, *shared fitness*), o qual incorpora uma penalização sobre f_i

conforme mais indivíduos estão situados no mesmo nicho. Desta maneira, durante o processo de seleção, o algoritmo tenderá a preferir indivíduos em regiões menos povoadas, mantendo certo grau de diversidade na população, ao mesmo tempo em que preserva máximos locais. Esta ideia é a essência do método conhecido como *fitness sharing*, e é descrita matematicamente abaixo [6, 9]:

$$f'_i = \frac{f_i}{\sum_{k=1}^{N_I} sh(d_{ik})} \quad (13)$$

onde N_I é o tamanho intermediário da população, d_{ik} é a distância (Euclidiana) entre o i -ésimo e o k -ésimo indivíduo e $sh(x)$ é chamada de função de compartilhamento, a qual realiza de fato o escalonamento do *fitness*. Uma função de compartilhamento bastante empregada é dada por:

$$sh(x) = \begin{cases} 1 - (x/\sigma)^\alpha, & \text{se } x \leq \sigma \\ 0, & \text{caso contrário} \end{cases} \quad (14)$$

Escolhendo-se $\alpha = 1$ (como é usual) [9] e $x = d_{ik}$, ela retorna o valor 1 caso os indivíduos sejam idênticos, o valor 0, caso a distância entre eles seja superior ao limite σ , e um valor intermediário que cresce com a diminuição de d_{ik} . Ou seja, quanto mais indivíduos estiverem a uma distância menor do que σ e quanto menor for essa distância em relação ao i -ésimo indivíduo, menor será seu *fitness* compartilhado f'_i .

A introdução do conceito de *niching* no algoritmo genético, porém, não se traduz somente em alterações no cálculo do *fitness*. Os operadores de permutação e seleção devem também ser modificados, de modo a favorecer a concretização do objetivo introduzido pelo *fitness sharing*.

Neste espírito, um estágio de restrição de cruzamento é acrescentado ao operador de recombinação, cuja descrição é dada a seguir:

- I. Caso um indivíduo seja escolhido para *crossover*, ele deve cruzar apenas com outro indivíduo do mesmo nicho (distância Euclidiana menor do que σ);
- II. Se não existir outro indivíduo da população no mesmo nicho, o cruzamento é feito, então, com outro indivíduo da população aleatoriamente escolhido.

A fim de preservar indivíduos de diferentes nichos, o método de seleção denominando SUS (do inglês, *Stochastic Universal Sampling*) pode ser utilizado [6]. Primeiro, cria-se a probabilidade de seleção de cada indivíduo calculando-se seu *fitness* relativo (a todos os *fitness* da população). Depois, define-se um passo aleatório μ pertencente a $[0, 1/L]$, onde

L é o número de indivíduos selecionados ao final (que deve ser igual ao tamanho original da população N).

Então, um laço é percorrido, indo de 1 a N_I , sendo N_I o tamanho da população após *crossover*. Para cada indivíduo, o passo μ é dado c vezes até superar o valor da probabilidade de seleção associada ao indivíduo em questão. O valor c indica o número de descendentes (cópias idênticas ao pai) que estarão presentes na próxima geração. Note que quanto maior o *fitness* do indivíduo, maior a chance de mais descendentes serem gerados. Porém, ao contrário do método de seleção por torneio, o SUS não é tão elitista, pois tende a manter um maior equilíbrio no número de descendentes entre os grupos dos melhores e piores indivíduos.

3.3 Clearing

O método de *clearing* pode ser empregado no lugar do *fitness sharing* resultando em uma abordagem um pouco mais agressiva. Ao invés do compartilhamento de recursos dentro de um nicho, todos os recursos são atribuídos ao(s) melhor(es) indivíduo(s), enquanto os demais indivíduos passam a ter *fitness* igual a zero. Essa ideia contribui para uma convergência mais rápida aos máximos locais, evitando manter muitos indivíduos explorando os mesmos nichos, ao contrário do *fitness sharing*.

Dada a matriz da população \mathbf{P} ordenada por linha de forma decrescente pelo vetor de *fitness* F , assim como seu tamanho intermediário N_I , o número de indivíduos suportados por um nicho κ e a distância σ , cria-se a rotina de *clearing*. Seu pseudo-código pode ser visto no Algoritmo 1.

Algorithm 1

```

1: procedure CLEARING( $F, \mathbf{P}, N_I, \kappa, \sigma$ )
2:    $D \leftarrow \text{dist}(\mathbf{P})$             $\triangleright$  matriz simétrica com distâncias par a par entre indivíduos
3:   for  $l = 1; l < N_I; l++$  do
4:      $C = \text{find}(D(l, 1:\text{end}) < \sigma)$     $\triangleright$  salva posição de indivíduos do mesmo nicho
5:     if  $F(l) \neq 0$  then  $F(C(\kappa:\text{end})) = 0$     $\triangleright$  zera fitness de indivíduos do nicho

```

A matriz D contém em seu elemento d_{ij} a distância Euclidiana do indivíduo i (isto é, da linha i na matriz \mathbf{P}) ao indivíduo j , que é idêntica à distância do indivíduo j ao i , de modo que $d_{ji} = d_{ij}$. Em seguida, há o laço que percorre a população a partir do indivíduo com maior *fitness*. A cada iteração, este indivíduo, chamado de referência, terá o mesmo índice das linhas de D . Dito isso, a matriz C salva os índices (linhas na matriz \mathbf{P}) dos indivíduos cuja distância à referência é menor do que σ (mesmo nicho). Caso a referência

não tenha tido seu *fitness* anulado previamente, os *fitness* dos outros indivíduos diferentes dos κ primeiros guardados em C têm seus *fitness* anulados. Observe que, por estarem em ordem decrescente, os indivíduos que terão seus *fitness* anulados nunca apresentam um *fitness* com valor superior ao de referência. Isso implica também que um indivíduo de índice i_1 nunca incapacitará um outro de índice $i_2 < i_1$, e que um indivíduo com *fitness* já anulado não poderá incapacitar outros indivíduos.

Por fim, sobrarão na população somente κ indivíduos de maior *fitness* em cada *nicho* e indivíduos de *fitness* anulados. Note que ao aplicar-se o SUS sobre esta população, indivíduos que tiveram seus *fitness* anulados não produzem descendentes.

O pseudo-código completo do algoritmo genético que incorpora o conceito de *clearing* pode ser visto abaixo:

Algorithm 2 Algoritmo Genético - *Clearing*

```

1: procedure MAIN( $M, K, N, \kappa, \sigma, \mu, \sigma_s^2, p_s, p_c$ )
2:    $[F, \mathbf{P}] \leftarrow \text{inicializar}(M, K, N)$   $\triangleright$  Gera população e fitness inicial associado
3:   for critério de parada não satisfeito do
4:      $[\mathbf{P}_{cross}, N_I] \leftarrow \text{crossover}(\mathbf{P}, N, \sigma, p_c)$   $\triangleright$  Nova população e tamanho
5:      $\mathbf{P}_{mut} \leftarrow \text{mutação}(\mathbf{P}_{cross}, N_I, p_s, \mu, \sigma_s^2)$ 
6:      $F' \leftarrow \text{fitness}(\mathbf{P}_{mut}, N_I, M, K)$ 
7:      $F'' \leftarrow \text{clearing}(F', \mathbf{P}_{mut}, N_I, \kappa, \sigma)$ 
8:      $[F''', \mathbf{P}_{sel}] \leftarrow \text{seleção}(F'', \mathbf{P}_{mut}, N_I, N)$ 

```

Nele, N é o tamanho original da população e todas as outras grandezas já foram definidas.

Concluídas as explicações sobre os modelos AR e ARMA, assim como a descrição do algoritmo genético que será utilizado para resolver o problema de otimização deste último (ver Algoritmo 2), apresentaremos na próxima seção características relevantes da série temporal utilizada no trabalho, bem como a divisão de dados aplicada para as etapas de treinamento e teste de ambos os modelos. As etapas serão explicadas de modo genérico para, depois, serem detalhadas nas exposições das implementações particulares do AR e do ARMA, referenciando sempre a teoria discutida nas Seções 2 e 3. Concomitantemente, os resultados obtidos para ambos os modelos das etapas de validação e teste serão relatados, analisados e, ao final, comparados entre si.

4 Metodologia e Resultados

4.1 Descrição da Série e Divisão de Dados

A série de vazões fluviais diárias (em m^3/s) utilizada neste projeto foi a da usina de Água Vermelha, considerando-se um período de 16 anos (de 1 de janeiro de 2000 a 31 de dezembro de 2015). Utilizou-se da validação cruzada (do inglês, *cross validation*) do tipo *holdout* para a separação dos dados, com o intuito de monitorar a generalização dos modelos [10]. Mais especificamente, os primeiros 5114 dias (1 de janeiro de 2000 a 31 de dezembro de 2013) constituíram os dados de treinamento e validação, os quais foram fornecidos ao previsor. Já os últimos 730 dias (01 de janeiro de 2014 a 31 de dezembro de 2015) foram usados na etapa de teste.

Essa série (dividida nos dois casos) pode ser observada nas Figuras 2 e 3, bem como alguns de seus dados relevantes mostrados na Tabela 1.

Tabela 1: Dados - Água Vermelha.

	Treinamento e validação	Teste
Média (m^3/s)	2059,1	1041,5
Desvio padrão (m^3/s)	1346,4	539,2
Máxima (m^3/s)	8641	2693
Mínima (m^3/s)	302	269

4.2 Implementação

A mesma metodologia foi empregada durante o projeto e avaliação dos modelos AR e ARMA, de modo a tornar verossímil a comparação de seus respectivos desempenhos. Por conseguinte, a implementação de ambos modelos foi separada em duas etapas: a etapa de validação e a de teste.

Na etapa de validação, os primeiros 14 anos foram divididos em duas partes: os dados de treinamento (01 de janeiro de 2000 até 31 de dezembro de 2009, totalizando 3653 dias) e os dados de validação (01 de janeiro de 2010 até 31 de dezembro de 2013, totalizando 1641 dias). Já na etapa de testes, foram utilizados 2 anos de informação, de 01 de janeiro de 2014 até 31 de dezembro de 2015, totalizando 730 dias.

Uma descrição geral de ambas as etapas pode ser vista abaixo:

1. Etapa de Validação

- 1.1. Sobre os dados de treinamento, para o modelo AR, encontra-se o conjunto de parâmetros ótimo para dado valor de atraso M (no caso do ARMA, para valores de M e K);
- 1.2. Calcula-se o EQM (erro quadrático médio) que o conjunto exhibe sobre os dados de validação. Para o modelo ARMA, faz-se o monitoramento do EQM de validação ao longo da execução do algoritmo genético que incorpora o conceito de *clearing* (ver Algoritmo 2);
- 1.3. O processo é repetido para outros valores de M pertencentes a um intervalo I_M (no modelo AR). Para o modelo ARMA, o processo é repetido para outras combinações de M e K ;
- 1.4. O M (ou o par M e K) que produziu o menor EQM (sobre os dados de validação) é então retreinado para o vetor de treinamento concatenado com o de validação (indo de 1 de janeiro de 2000 a 31 de dezembro de 2013, totalizando 5294 dias). O conjunto de parâmetros resultante é levado para a Etapa de Teste.

2. Etapa de Teste

- 2.1. O conjunto de parâmetros final é usado para prever todas as vazões presentes nos dados de teste (de 1 de janeiro de 2014 a 31 de dezembro de 2015);
- 2.2. Baseado nas saídas geradas, calcula-se o respectivo EQM e EAM (erro absoluto médio). Além disso, é plotado o gráfico da série original do vetor de teste sobre as vazões previstas pelo modelo, para uma comparação visual. Todos estes três elementos serão utilizados para a comparação da eficiência entre os modelos AR e ARMA.

A descrição específica da implementação de cada modelo será detalhada nas Seções 4.3 e 4.4.

4.2.1 Motivação para a escolha dos atrasos M e K

O intervalo de valores para M foi definido através da função de autocorrelação parcial aplicada sobre os dados de treinamento e validação concatenados. A autocorrelação parcial encontra a correlação dos resíduos (chamados assim pois são mantidos após a remoção dos efeitos já explicados pelos primeiros atrasos) com o próximo valor de atraso, removendo correlações entre este e atrasos passados.

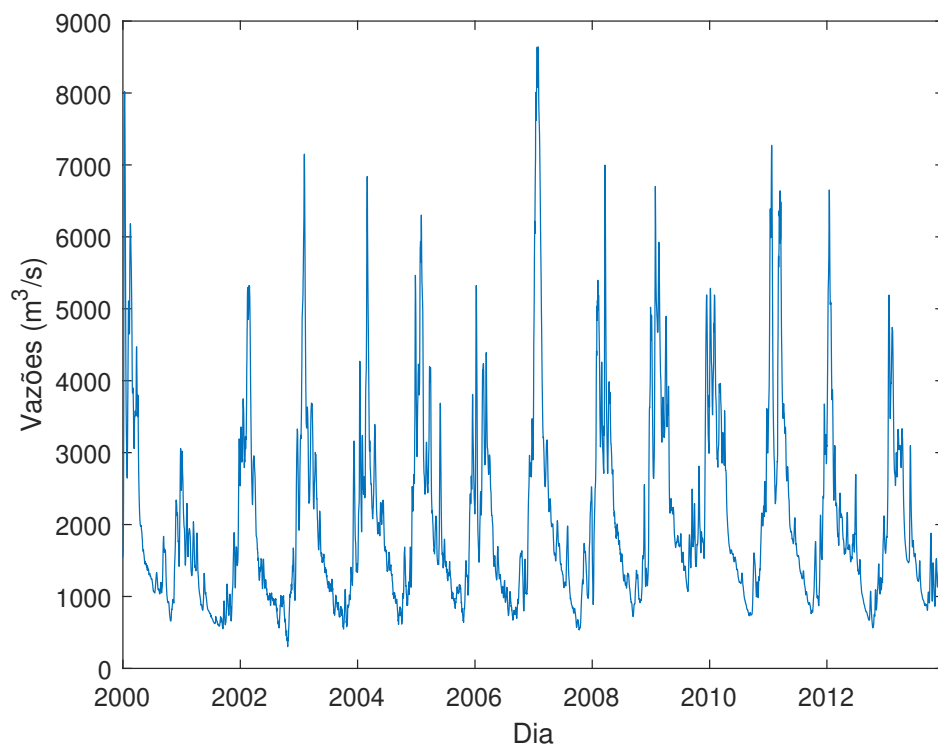


Figura 2: Série de Vazões de Água Vermelha - Dados de treinamento.

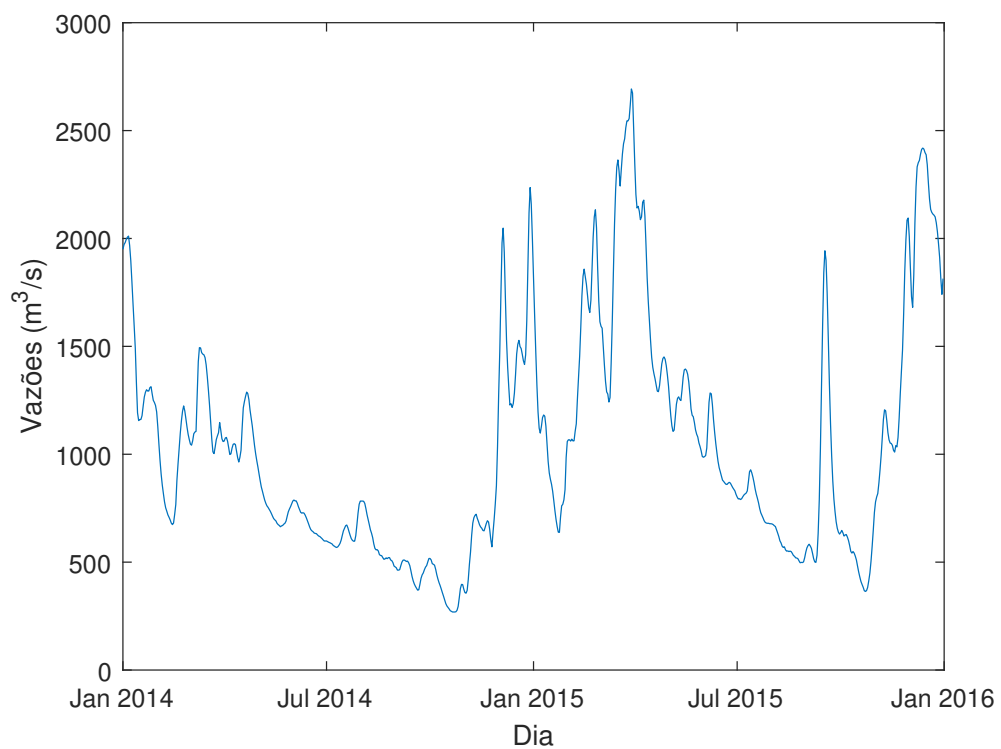


Figura 3: Série de Vazões de Água Vermelha - Dados de teste.

Assim, a autocorrelação parcial descreve somente a relação direta entre duas observações da série afastadas por m instantes de tempo. Dado que o modelo AR incorpora justamente valores da própria série, isso sugere que não haverá correlação para atrasos acima de M_{max} .

O gráfico da função de autocorrelação parcial, considerando as vazões no período compreendido pelos conjuntos de treinamento e validação da série de Água Vermelha, pode ser visto na Figura 4.

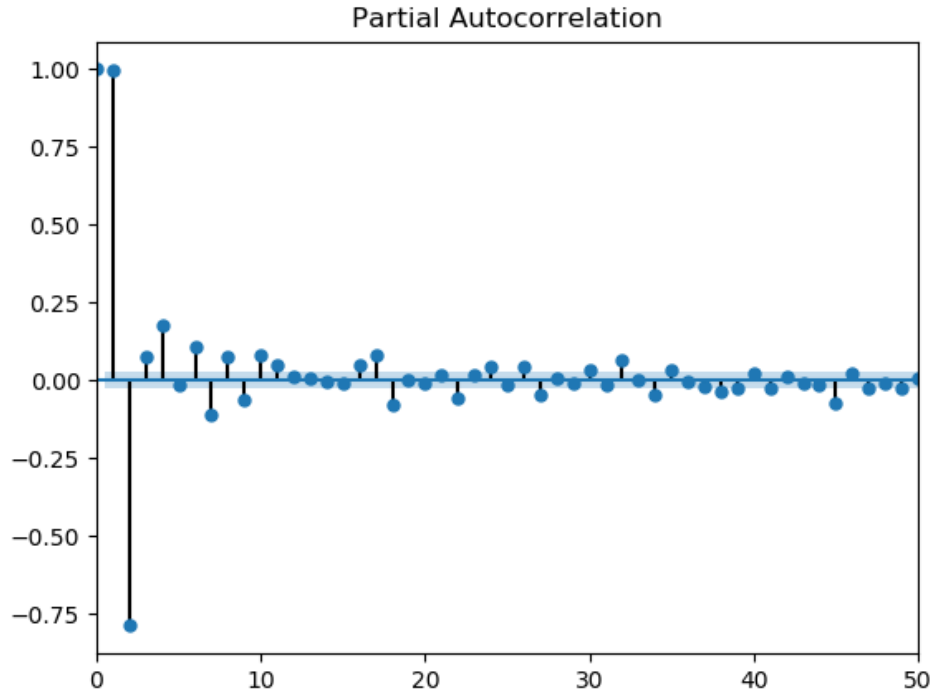


Figura 4: Autocorrelação parcial dos dados de treinamento e validação.

A função que calcula a autocorrelação parcial de uma série apresenta um intervalo de confiança, o qual se situa fora da região azul na Figura 4. Analisando-a, conclui-se que o maior intervalo que considera uma maioria de pontos na região de confiança e de autocorrelação significativa é o que começa em 1 e termina em $M_{max} = 30$.

Analogamente, o intervalo de valores para K foi definido a partir da função de autocorrelação sobre os dados de treinamento e validação.

A função de autocorrelação descreve quão bem o valor presente da série é relacionado com seus valores passados. Como para o modelo MA o valor no instante n é uma combinação linear de resíduos (erros ou *ruídos brancos*) e nenhum valor atrasado da própria série pode diretamente explicar sua saída atual, não existe aqui o caráter sazonal das correlações (isto é, a volta da existência de correlações com valores altos para grandes

atrasos) entre duas observações da própria série. Deste modo, a autocorrelação descreverá somente a relação direta dos resíduos afastados por k instantes de tempo. Isso sugere que não haverá correlação de valor significativo para um atraso maior do que K_{max} , exatamente o desejado.

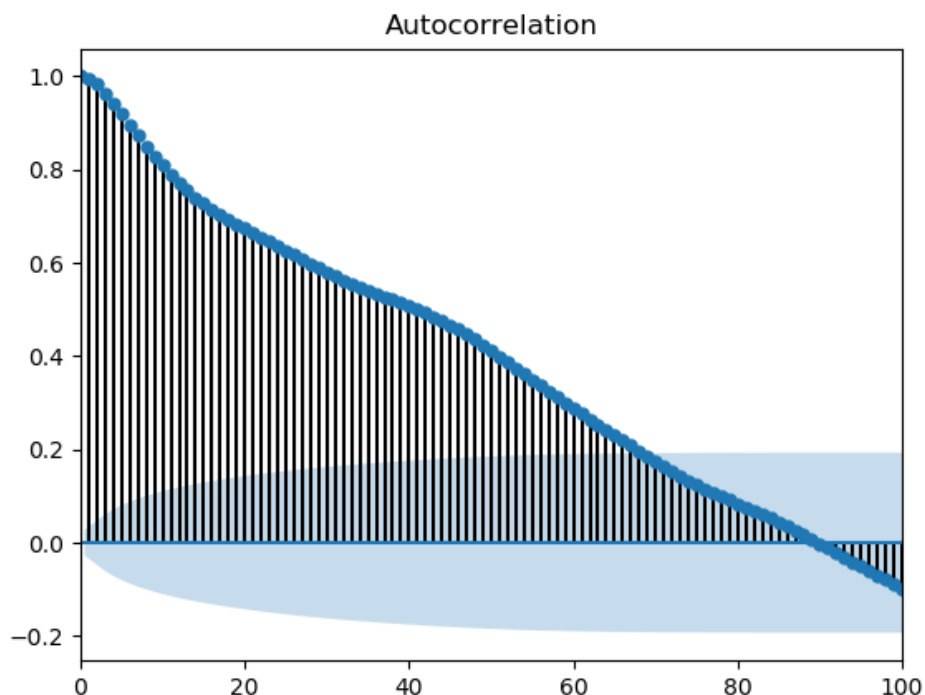


Figura 5: Autocorrelação dos dados de treinamento e validação.

Observa-se na Figura 5 que a partir do atraso igual a 60, deixa-se o intervalo de confiança e os valores da autocorrelação se tornam muito baixos.

A princípio, o intervalo de consideração iria até o valor 60. Porém, após realizados testes com o modelo ARMA, conclui-se que, para diversos valores de M pertencentes a I_M , um $K = 60$ sempre levava a saídas divergentes. Com efeito, o sistema começava a apresentar divergência para valores de $K = 40$. Por isso, decidimos reduzir o intervalo de consideração para K indo de 1 até $K_{max} = 40$.

4.3 Modelo AR

Para a etapa de validação (ver Seção 4.2) deseja-se encontrar o valor de atraso M dentro de I_M (ver Seção 4.2.1) que leve ao menor EQM sobre os dados de validação.

Como o modelo AR apresenta solução fechada para cada número de atrasos escolhido, foi necessário apenas um laço externo que percorresse o intervalo I_M . Para dado tamanho

de atraso, resolve-se um sistema descrito na forma matricial por (7). Note que as autocorrelações em (6) – $r(0), r(1), \dots, r(M)$ –, pertencem todas ao vetor de treinamento.

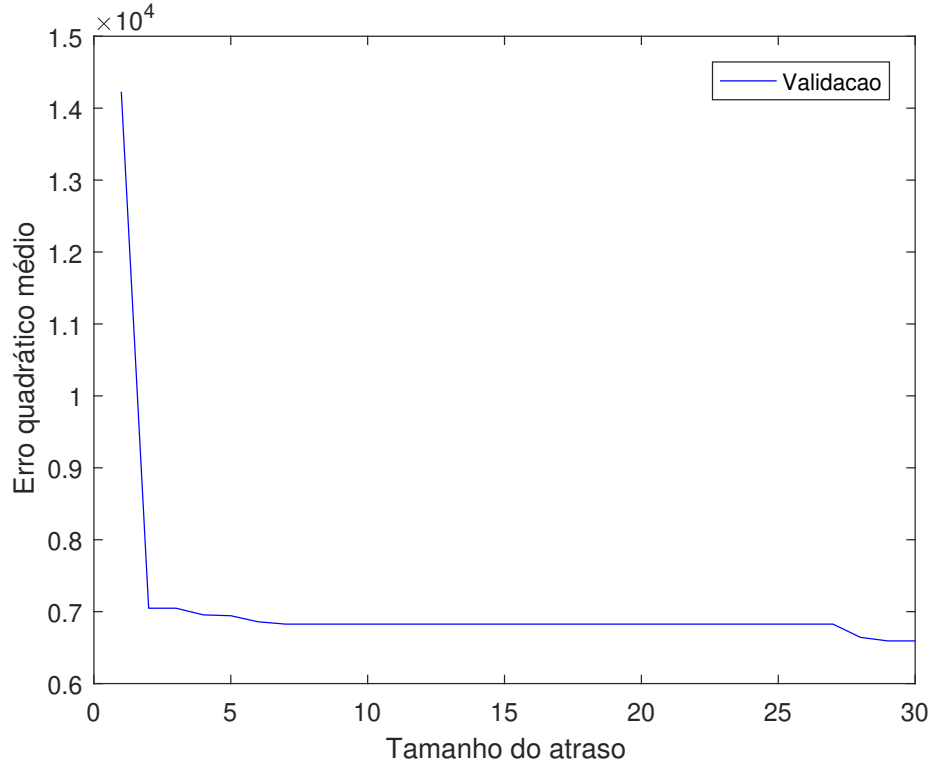


Figura 6: Erro quadrático médio por tamanho do atraso - AR.

Tabela 2: Resultados - Modelo AR.

Erro Quadrático Médio $(m^3/s)^2$	1088,6
Erro Absoluto Médio (m^3/s)	19,8

O vetor de parâmetros encontrados \mathbf{w} é utilizado para prever as vazões diárias dos dados de validação, como mostra a Equação (1). As previsões, por sua vez, são usadas para o cálculo do EQM.

Este erro, associado a certo tamanho de atraso, é salvo em um vetor. O processo, então, é repetido para os outros valores de M em I_M . Com isso, podemos gerar um gráfico do EQM em função do número de atrasos utilizados no modelo AR, o qual pode ser visto na Figura 6.

Nele, nota-se que existe pouca variação no EQM de validação em função do atraso. Com exceção do caso em que $M = 1$, a diferença entre os EQMs obtidos é praticamente inexistente. De qualquer forma, para $M \in I_M$, o menor EQM, de $6593 (m^3/s)^2$, está associado ao atraso de $K = 29$.

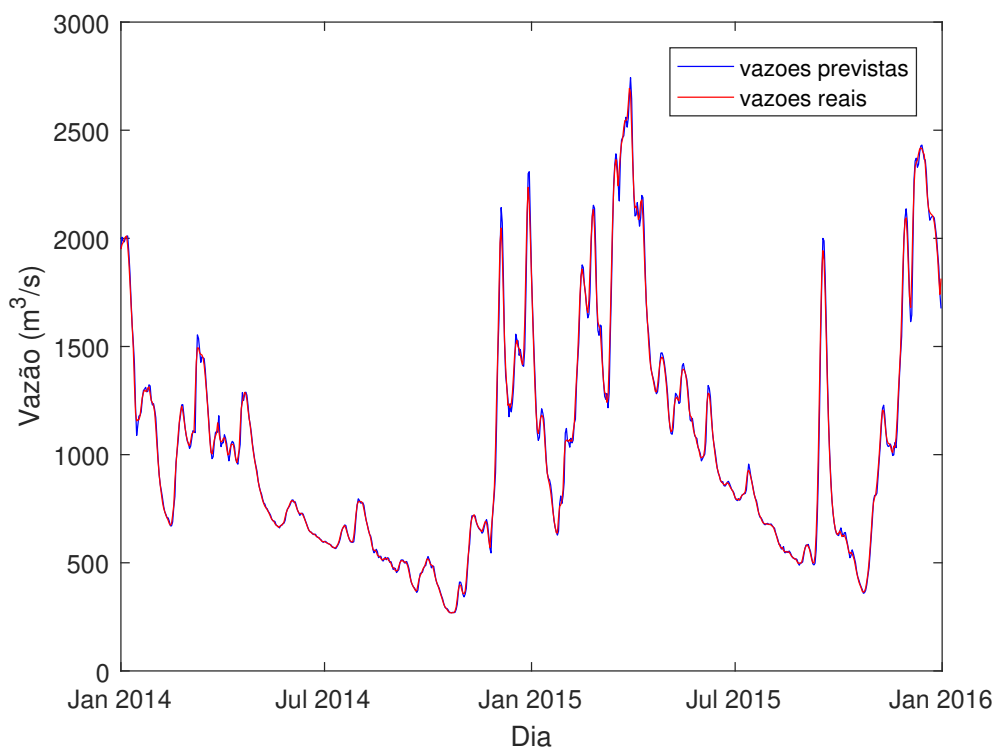


Figura 7: Vazões previstas e vazões reais - AR

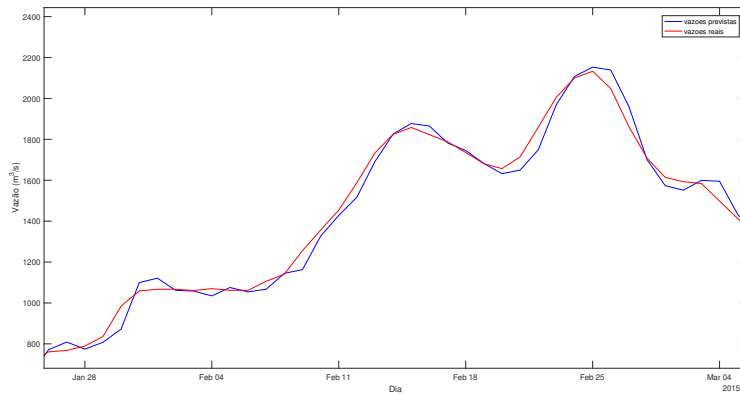


Figura 8: Vazões previstas e vazões reais com *zoom* - AR

Sendo assim, utiliza-se este número de entradas no preditor AR em um novo treinamento, considerando agora os dados de validação e treinamento concatenados.

Utilizando o conjunto de parâmetros ótimo obtido neste último treinamento, obtém-se as saídas do modelo AR (de acordo com a Equação (1)) para o conjunto de teste e calcula-se os valores de EQM e EAM, os quais são mostrados na Tabela 2.

Por sua vez, o gráfico que sobrepõe as vazões originais de Água Vermelha com aquelas previstas na etapa de teste pelo modelo AR é mostrado na Figura 7. Com o objetivo de

melhorar a visualização, o gráfico de previsões em um período agora de 1 mês é mostrado na Figura 8.

É possível perceber na Figura 8 que as curvas de vazões estão deslocadas. Este atraso de 1 dia aparente é intrínseco ao modelo de previsões empregado.

Em suma, constatamos que o modelo AR apresentou um bom resultado, dado que a curva de previsões acompanha a série de vazões original e o EAM foi de apenas $19,8 \text{ m}^3/\text{s}$.

4.4 Modelo ARMA

Para a etapa de validação, deseja-se encontrar primeiramente o conjunto ótimo de parâmetros para cada uma das combinações de atrasos M e K escolhidas.

Os valores de M e K (ver Seção 4.2.1) aqui considerados foram:

- $\mathbf{M} = [1, 5, 15, 30]$
- $\mathbf{K} = [1, 5, 10, 20, 40]$

Analogamente ao modelo AR, foram testadas cada uma das 20 combinações de K e M possíveis. Dada uma combinação de atrasos, cria-se inicialmente uma população de 100 indivíduos, sendo que cada elemento é gerado a partir de uma distribuição Gaussiana padrão (média nula e variância unitária). Cada indivíduo é representado por um vetor linha de tamanho $G = M + K + 1$ (sendo o último termo o coeficiente independente).

Inicializada a população, aplica-se o *crossover* aritmético (ver Seção 3.1.1) com o conceito de *niching* (ver seção 3.2). Adotamos uma probabilidade de crossover $p_c = 80\%$ e um raio $\sigma = 0,5$.

Depois, é aplicada a *mutação gaussiana* (ver Seção 3.1.2). A probabilidade de mutação é de $p_s = 20\%$, enquanto a distribuição normal adicionada tem média $\mu = 0$ e variância de $\sigma_m^2 = 0.5$.

O próximo passo é calcular ambos os *fitness* (inverso do EQM) de cada indivíduo da população: um sobre os dados de treinamento e outro sobre os dados de validação. Isso é feito gerando-se as saídas do ARMA (ver Equação (9)) sobre os respectivos dados. Ordena-se, então, a população em relação ao vetor coluna *fitness* de treinamento e ao vetor coluna *fitness* de validação (em duas matrizes separados).

Sobre a população com *fitness* associado aos dados de treinamento é feito o procedimento de *clearing* (ver Seção 3.3). O número de indivíduos permitido por nicho é de $\kappa = 1$.

Novamente, esta população é ordenada por *fitness* de forma decrescente. Em seguida, é aplicado o operador de seleção SUS (ver Seção 3.2).

Caso o melhor *fitness* (sobre os dados de treinamento) da geração atual seja menor que o maior *fitness* da geração passada, troca-se o pior indivíduo da geração atual pelo melhor indivíduo da passada. Essa estratégia é mencionada na Seção 3.1.3, apesar de que aqui não se utiliza a seleção por torneio.

Tabela 3: Coeficientes Relevantes

Coeficientes Gerais	
Dados de treinamento	3653
Dados de validação	1461
Dados de teste	730
Tamanho da população	100 / 150
Número de Gerações	150 / 300
Atraso das amostras M	[1, 5, 15, 30]
Atraso dos erros K	[1, 5, 10, 20, 40]
Passo de amostra (l)	1
Passo de erro (j)	1
Coeficientes de <i>crossover</i>	
Raio do <i>nicho</i> σ	0,5
Probabilidade de <i>crossover</i> p_c	0,8
Coeficientes de mutação	
Média μ	0,5
Variância σ_m^2	0,5
Probabilidade de mutação p_s	0,8
Coeficientes de <i>clearing</i>	
Indivíduos por nicho κ	1

Por fim, compara-se agora o indivíduo com maior *fitness* de validação da geração atual com o da geração passada. Se houve uma melhora, salva-se o novo indivíduo no lugar do anterior em uma variável de memória. Caso contrário, continua-se com a melhor solução

obtida até então.

Todos os parâmetros relevantes ao código que foram declarados anteriormente estão registrados na Tabela 3.

O processo é repetido por 150 gerações, e produz o menor EQM (o maior *fitness*) associado ao vetor de validação para dada combinação de M e K . Como o modelo ARMA não possui solução fechada, o processo descrito acima é repetido 5 vezes, e o EQM final considerado é a média entre todos os EQMs produzidos para cada combinação de atrasos. A Tabela 4 apresenta os valores médios dos EQMs obtidos nos experimentos realizados. A princípio, nota-se uma grande diferença entre os valores (indo da ordem de 10^3 até 10^5). Por essa razão, decide-se desconsiderar os valores de ordem 10^5 , os quais são divergentes, para plotar o histograma da Figura 9.

Tabela 4: Resultados da Validação - Modelo ARMA

M	K	EQM $(m^3/s)^2$
1	1	$9720,1 \pm 0,2$
1	5	$7503,1 \pm 53,8$
1	10	$7797,2 \pm 117,6$
1	20	$8682,9 \pm 657,9$
1	40	113510 ± 147340
5	1	$6894,5 \pm 47,2$
5	5	$7308,2 \pm 372,3$
5	10	$7811,6 \pm 802,2$
5	20	14449 ± 4668
5	40	194420 ± 222210
15	1	$7937,1 \pm 578,8$
15	5	$9481,4 \pm 1554,7$
15	10	10351 ± 2774
15	20	17124 ± 6728
15	40	166600 ± 170740
30	1	10563 ± 1821
30	5	10065 ± 1235
30	10	13334 ± 3737
30	20	17248 ± 2820
30	40	230830 ± 280120

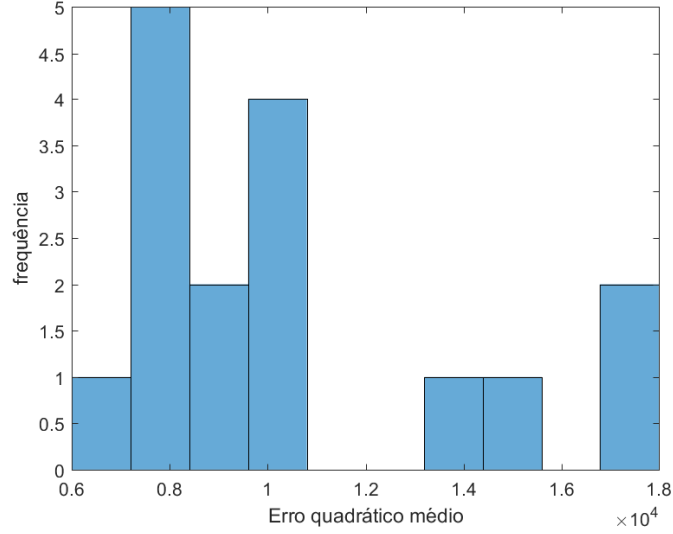


Figura 9: Histograma do erro quadrático médio - ARMA

Podemos observar que a variedade de valores de EQM é muito maior para o ARMA do que aquela vista no modelo AR. Isso não apenas se deve à maior exploração do espaço de busca ao utilizar-se uma meta-heurística (dado que não se conhece a solução ótima), mas também à possível instabilidade do modelo ARMA dependendo da combinação de atrasos M e K escolhida.

Simultaneamente, pode-se observar na Tabela 4 que um aumento no número de atrasos K leva, no geral, a um maior EQM, apresentando divergência (EQM na ordem de 10^5) em todos os casos em que $K = 40$ (inclusive, os desvios padrões apresentam mesma ordem que suas respectivas médias nestes casos). Um aumento no valor de M implica também, embora menos drasticamente, em um aumento do EQM. A exceção ocorre na combinação de atrasos ótima, que corresponde a: $M = 5$ e $K = 1$.

Utilizando este número de atrasos “ótimo”, o modelo ARMA é retreinado exatamente como descrito anteriormente. Contudo, agora os dados de treinamento e validação são concatenados e o número de gerações assim como o tamanho da população aumentam para 300 e 150, respectivamente. O conjunto de parâmetros ótimo ao final produz todas as saídas (de acordo com a Equação (9)) para o conjunto de teste. A partir delas são calculadas os valores de EQM e EAM, registrados na Tabela 5.

Tabela 5: Resultados do Teste - Modelo ARMA

Erro Quadrático Médio $(m^3/s)^2$	1056,3
Erro Absoluto Médio (m^3/s)	19,7

Adicionalmente, o gráfico que sobrepõe as vazões originais de Água Vermelha com

aquelas previstas na etapa de teste pelo modelo ARMA é mostrado na Figura 10. Para melhorar a visualização, o mesmo gráfico é plotado na Figura 11 considerando o período de 1 mês utilizado para o modelo AR.

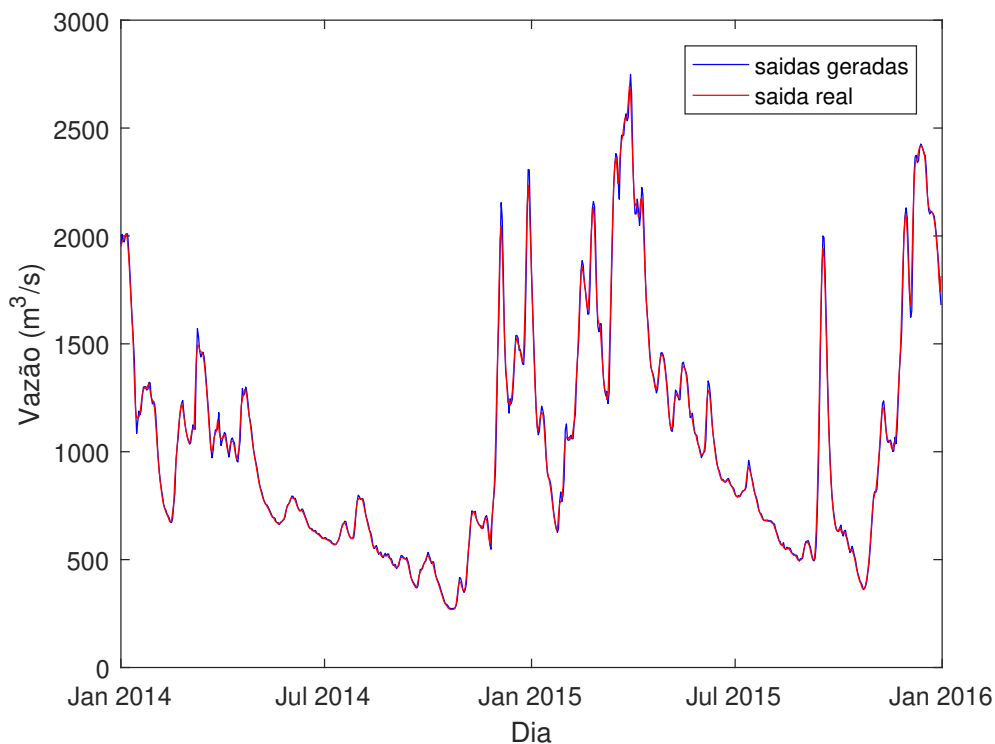


Figura 10: Vazões previstas e vazões reais - ARMA

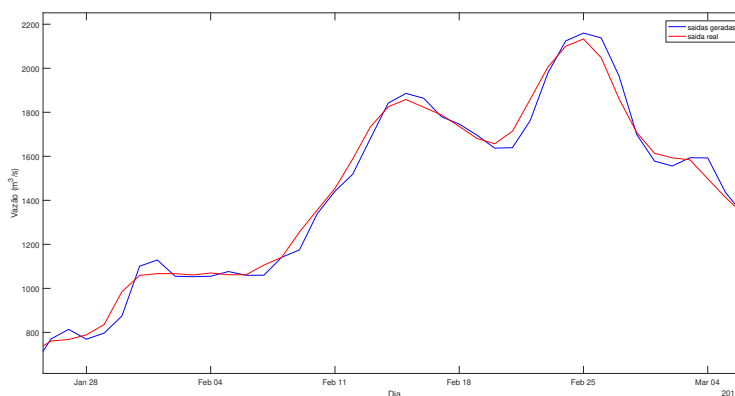


Figura 11: Vazões previstas e vazões reais com *zoom* - ARMA

Pode-se notar que as curvas de vazões, assim como no modelo AR, estão deslocadas. Novamente, este atraso de 1 dia aparente é intrínseco ao modelo de previsões empregado.

É possível constatar que o modelo ARMA apresentou um resultado satisfatório, dado que a curva de previsões acompanha a série de vazões original e o EAM foi de $19,7 \text{ m}^3/\text{s}$.

4.5 Comparação entre modelos

Comparando-se ambos os modelos, observa-se um resultado praticamente idêntico: uma diferença de $32,3 (\text{m}^3/\text{s})^2$ no EQM e de $0,1 \text{m}^3/\text{s}$ no EAM sobre os dados de teste. Curiosamente, o tamanho de atraso utilizado para o AR foi de $M = 29$, enquanto no modelo ARMA os atrasos utilizados foram menores, com $M = 5$ e $K = 1$.

Isso indica que o acréscimo da informação do erro de previsão passado $\eta(n - 1)$ foi decisivo para que o modelo ARMA pudesse estimar com maior precisão o valor futuro da série. Ou seja, para atingir um EAM próximo de $19,8 \text{m}^3/\text{s}$, o modelo AR precisou utilizar $M = 29$ atrasos, enquanto o modelo ARMA precisou apenas de $M = 5$ atrasos em conjunto com o valor de $\eta(n - 1)$.

5 Conclusões

Até esta etapa do projeto, foram estudados e implementados os modelos lineares AR e ARMA para a previsão de uma série de vazões, a de Água Vermelha, cujos dados foram divididos utilizando-se da validação cruzada do tipo *holdout*.

Ademais, para otimizar o modelo ARMA, foram estudadas meta-heurísticas inspiradas no processo evolutivo, as quais trabalham com populações de soluções candidatas e utilizam os processos de reprodução, mutação e seleção para conduzir a busca. Em particular, optou-se pelo uso do algoritmo genético que incorpora o conceito de *clearing*, devido à sua grande capacidade de exploração da região de busca e eficiência em preservar seus máximos locais.

Ao final, os desempenhos de ambos os métodos, baseados nos EQMs cometidos na previsão dos dados de teste, foram analisados e comparados entre si. No fundo, foi constatada uma equivalência no desempenho final obtido pelas melhores configurações dos modelos AR e ARMA. Por um lado, o modelo ARMA precisou de menos atrasos (da série e do erro de previsão) para chegar àquele desempenho; por outro lado, seu processo de treinamento é consideravelmente mais custoso que o do modelo AR.

Com isso, concluímos todas as metas previstas para a primeira metade desta iniciação científica.

A fim de dar continuidade ao projeto, pretende-se implementar os modelos AR e ARMA considerando outras séries de vazões. Além disso, deseja-se utilizar métodos não-lineares para a previsão de séries, como, por exemplo, redes neurais artificiais. Dentre as várias redes existentes, planejamos considerar a rede perceptron de múltiplas camadas (MLP, do inglês *multilayer perceptron*), como representante da classe de modelos

feedforward, e, também, dois modelos recorrentes: as LSTM (*long short-term memory*) [11] e as GRUs (*gated recurrent units*) [12].

Referências

- [1] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley, 2015.
- [2] S. Haykin, *Adaptive filter theory*, 5th ed. Prentice Hall, 2013.
- [3] T. M. Bartlett, “Modelagem de séries temporais não-estacionárias através de um modelo arma multimomental baseado em misturas de componentes normais,” Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2018.
- [4] P. P. Vaidyanathan, *The Theory of Linear Prediction*. Morgan and Claypool Publishers, 2008.
- [5] H. V. Siqueira, “Previsão de séries de vazões com redes neurais artificiais e modelos lineares ajustados por algoritmos bio-inspirados,” Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2009.
- [6] L. Boccato, “Aplicação de computação natural ao problema de estimação de direção de chegada,” Dissertação de Mestrado, Faculdade de Engenharia Elétrica e de Computação, UNICAMP, 2010.
- [7] T. Bäck, D. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. Bristol, UK: Institute of Physics Publishing, 2000.
- [8] J. H. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press., 1975.
- [9] e. R. J. Goldberg, D. E., “Genetic algorithms with sharing for multimodal function optimization,” *2nd Int. Conf. on Genetic Algorithms*, 1987.
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cambridge, United Kingdom: Springer, 2011.
- [11] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “Lstm: A search space odyssey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

- [12] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv:1412.3555 [cs.NE]*, 2014.

Nº /

ANEXO 02 À RESOLUÇÃO CG/FEEC 01
DISCIPLINAS EA006, EE013, EE014, EE015 e EE016
Formulário de Orientador – Encaminhamento do Relatório

IDENTIFICAÇÃO

SIGLA DISCIPLINA: EE015

ALUNO: RA 169408

Nome: Daniel da Costa Nunes Resende Neto

ORIENTADOR: Prof: Levy Boccato **Ramal:** 13800

TÍTULO DO TRABALHO: Estudo e Aplicação de Modelos de Previsão no Contexto de Séries de Vazões de Rios

APRECIÇÃO DO ORIENTADOR: Desempenho do aluno (utilize o verso, se necessário)

O aluno Daniel Neto possui um histórico escolar muito bom. Durante este semestre, o aluno realizou de forma plena todas as atividades previstas em sua iniciação científica. Na etapa de estudos e revisão bibliográfica, o aluno reuniu-se periodicamente com o orientador para discutir os temas ligados ao problema de predição de séries temporais e aos modelos clássicos (AR, MA e ARMA). O aluno, em um curto período, atingiu um domínio bastante amplo de conceitos fundamentais de áreas relacionadas ao problema, como processos estocásticos, aprendizado de máquina e computação evolutiva. Esta última temática foi abordada tendo em vista o treinamento do modelo ARMA.

Em todo o período, o aluno demonstrou ter uma vocação natural à pesquisa e uma motivação constante nos estudos.

Ótimo (X) Bom () Regular () Fraco ()

APRECIÇÃO DO ORIENTADOR: Relatório do aluno (utilize o verso, se necessário)

O relatório entregue pelo aluno resume todo o conhecimento adquirido neste semestre com respeito aos fundamentos do problema de predição de séries temporais, aos modelos lineares clássicos existentes (AR, MA e ARMA) e à metodologia científica para treinamento, validação e teste (além da seleção de hiperparâmetros). Nota-se, claramente, a fluência que o aluno desenvolveu nestes temas, expondo com clareza todos os conceitos cobertos em seu estudo. Os resultados obtidos foram devidamente apresentados e analisados, servindo como confirmação do bom andamento da pesquisa.

Esta primeira parte de sua iniciação, portanto, foi concluída com êxito, sendo o relatório um registro de seu desenvolvimento na temática, mostrando também o seu aprendizado com

respeito a atividades de pesquisa.

Ótimo (X) Bom () Regular () Fraco ()

Não será recebido após o prazo estabelecido.

Nota: 10,0 Frequência: 100% Orientador:



Prof. Levy Boccato
DCA - FEEC - UNICAMP
Matrícula 307343